

IN THE CLAIMS:

The text of all pending claims, (including withdrawn claims) is set forth below. Cancelled and not entered claims are indicated with claim number and status only. The claims as listed below show added text with underlining and deleted text with ~~strikethrough~~. The status of each claim is indicated with one of (original), (currently amended), (cancelled), (withdrawn), (new), (previously presented), or (not entered).

Please claims 12, 23, 34, and 40-42 in accordance with the following:

1.-7. (CANCELED)

8. (PREVIOUSLY PRESENTED) An apparatus according to claim 12, wherein a plurality of storage units respectively corresponding to a plurality of executors of said subroutine are prepared; and

said collecting section specifies said executor of said subroutine and stores said profile of said subroutine corresponding to said specified executor in said storage unit.

9. (ORIGINAL) An apparatus according to claim 8, wherein said collecting section individually stores profiles of a plurality of subroutines corresponding to a specified executor in said storage unit.

10. (ORIGINAL) An apparatus according to claim 9, wherein said collecting section individually stores a first profile of a subroutine called by a main routine, and a second profile of the subroutine called by another subroutine, in said storage unit.

11. (ORIGINAL) An apparatus according to claim 10, wherein said collecting section stores said second profile and calling relationship information relating to said second profile, said calling relationship information indicating a relationship between said other subroutine and said called subroutine, in said storage unit.

12. (CURRENTLY AMENDED) An apparatus for collecting a profile of a subroutine included in a program, comprising:
a storage unit storing a profile;
an analyzing section, when an interrupt is generated by execution of a branch instruction during execution of said program[[,]]:

obtaining a branch source address and a branch destination address from a source of said interrupt, and

identifying a type of said branch instruction by obtaining [[a]] an instruction code from said branch source address and decoding said instruction code; and

a collecting section;

obtaining said branch source address, said branch destination address, and [[a]] an identified result from said analyzing section when the identified instruction is a calling instruction or a return instruction of said subroutine; and

when said identified result is said calling instruction, storing said branch destination address as a subroutine address corresponding to said calling instruction and a calling time of said subroutine corresponding to said calling instruction in said storage unit; and,

when said identified result is said return instruction,

obtaining a return time of said subroutine corresponding to said return instruction,

calculating a execution time of said subroutine based on said obtained return time and said calling time, and

storing a cumulative value of said execution time as said profile in correspondence with said branch destination address in said storage unit~~[[a]]~~, and

when the identified branch instruction is neither a calling instruction nor a return instruction, said interrupt is terminated.

13. (PREVIOUSLY PRESENTED) An apparatus according to claim 12, wherein said collecting section stores times of calling of said subroutine corresponding to said branch destination address as said profile in said storage unit.

14. (ORIGINAL) An apparatus according to claim 12, wherein said collecting section obtains an overhead of said subroutine as said profile and stores said overhead in said storage unit.

15. (PREVIOUSLY PRESENTED) An apparatus according to claim 13, wherein said collecting section, when said identified result is said calling instruction, stores an identifier of an executor of said subroutine corresponding to said calling instruction and said branch destination address in said storage unit.

16. (PREVIOUSLY PRESENTED) An apparatus according to claim 12, wherein said collecting section, when said identified result is said calling instruction and said branch source address and said branch destination address are addresses of said subroutines, stores said branch source address and branch destination address as calling relationship information indicating a calling source subroutine and a calling destination subroutine in said storage unit, and stores at least one of the cumulative execution time and the times of calling in said calling destination subroutine in the call source subroutine, as said profile corresponding to said calling relationship information, in said storage unit.

17. (PREVIOUSLY PRESENTED) An apparatus according to claim 12, further comprising a setting section setting an execution environment of a source of said interrupt so as to generate said interrupt when said branch instruction is executed during the execution of said program.

18. (CANCELED)

19. (PREVIOUSLY PRESENTED) A computer readable medium according to claim 23, wherein said computer further executes the process by:

specifying an executor of said subroutine; and

storing said profile in a storage unit corresponding to said specified executor within a plurality of storage unit provided with each executor.

20. (PREVIOUSLY PRESENTED) A computer readable medium according to claim 19, wherein said computer further executes the process by storing, when a specified executor executes a plurality of subroutines, profiles of said plurality of subroutines in said storage unit corresponding to said specified executor.

21. (PREVIOUSLY PRESENTED) A computer readable medium according to claim 20, wherein said computer further executes the process of storing a first profile of said subroutine called by a main routine and a second profile of said subroutine called by another subroutine in said corresponding storage unit, regarding to each of said subroutines.

22. (PREVIOUSLY PRESENTED) A computer readable medium according to claim

21, wherein said computer further executes the process by storing said second profile and calling relationship information relating to said second profile, said calling relationship information indicating a relationship between said other subroutine and said subroutine.

23. (CURRENTLY AMENDED) A computer readable medium storing a program for a computer executing a process for collecting a profile of a subroutine included in a target program, by:

obtaining a branch source address and a branch destination address from a source of an interrupt when said interrupt is generated by execution of a branch instruction during execution of the target program;

identifying a type of said branch instruction by obtaining an instruction code from said branch source address and decoding said instruction code;

storing said branch destination address as a subroutine address corresponding to said calling instruction and a calling time of said subroutine corresponding to said calling instruction in a storage unit when said identified result is said calling instruction; and

when said identified result is said return instruction,

obtaining a return time of said subroutine corresponding to said return instruction,

calculating an execution time of said subroutine based on said obtained return time and said calling time, and

storing a cumulative value of said execution time as said profile in correspondence with said branch destination address in said storage unit, and

when the identified branch instruction is neither a calling instruction nor a return instruction, said interrupt is terminated.

24. (PREVIOUSLY PRESENTED) A computer readable medium according to claim 23, wherein said computer further executes the process by storing times of calling of said subroutine corresponding to said branch destination address as said profile in said storage unit.

25. (PREVIOUSLY PRESENTED) A computer readable medium according to claim 23, wherein said program further executes the process by storing an overhead of said subroutine as said profile in said storage unit.

26. (PREVIOUSLY PRESENTED) A computer readable medium according to claim 23, wherein said computer further executes the process by storing, when said identified result is

said calling instruction, an identifier of an executor of said subroutine corresponding to said calling instruction and said branch destination address in said storage unit.

27. (PREVIOUSLY PRESENTED) A computer readable medium according to claim 23, wherein said computer further executes the process when said identified result is said calling instruction and said branch source address and said branch destination address are addresses of said subroutines, storing said branch source address and branch destination address as calling relationship information indicating a calling source subroutine, and a calling destination subroutine in said storage unit; and storing at least one of the cumulative execution time and the times of calling in said calling destination subroutine in the call source subroutine, as said profile corresponding to said calling relationship information, in said storage unit.

28. (PREVIOUSLY PRESENTED) A computer readable medium according to claim 23, wherein said computer further executes the process by setting an execution environment of a source of said interrupt so as to generate said interrupt when said branch instruction is executed during the execution of said program.

29. (CANCELED)

30. (PREVIOUSLY PRESENTED) A method according to claim 34, further comprising:

- specifying an executor of the subroutine; and
- storing the profile in a storage unit corresponding to the specified executor within a plurality of storage unit provided with each executor.

31. (PREVIOUSLY PRESENTED) A method according to claim 30, further comprising storing, when a specified executor executes a plurality of subroutines, profiles of the plurality of subroutines in the storage unit corresponding to the specified executor.

32. (PREVIOUSLY PRESENTED) A method according to claim 31, further comprising storing a first profile of the subroutine called out by a main routine and a second profile of the subroutine called out by another subroutine in the corresponding storage unit, regarding to each of the subroutines.

33. (PREVIOUSLY PRESENTED) A method according to claim 32, further comprising storing the second profile and calling relationship information relating to the second profile, the calling relationship information indicating a relationship between the other subroutine and the subroutine.

34. (CURRENTLY AMENDED) A method for collecting a profile of a subroutine included in a program, comprising:

- obtaining a branch source address and a branch destination address from a source of an interrupt when the interrupt is generated by execution of a branch instruction during execution of said program;

- identifying a type of the branch instruction by obtaining [[a]] an instruction code from the branch source address and decoding the instruction code;

- storing the branch destination address as a subroutine address corresponding to the calling instruction and a calling time of the subroutine corresponding to the calling instruction in a storage unit when the identified result is the calling instruction; and

- when the identified result is the return instruction,

 - obtaining a return time of the subroutine corresponding to the return instruction,

 - calculating a execution time of the subroutine based on the obtained return time and the calling time, and

- storing a cumulative value of the execution time as the profile in correspondence with the branch destination address in the storage unit~~[[:]]~~, and

- when the identified branch instruction is neither a calling instruction nor a return instruction, said interrupt is terminated.

35. (PREVIOUSLY PRESENTED) A method according to claim 34, further comprising storing times of calling of the subroutine corresponding to the branch destination address as the profile in said storage unit.

36. (PREVIOUSLY PRESENTED) A method according to claim 34, further comprising storing an overhead of the subroutine as the profile in the storage unit.

37. (PREVIOUSLY PRESENTED) A method according to claim 34, further comprising storing, when the identified result is the calling instruction, an identifier of an executor of the subroutine corresponding to the calling instruction and the branch destination address in

the storage unit.

38. (PREVIOUSLY PRESENTED) A method according to claim 34, further comprising, when the identified result is the calling instruction and the branch source address and the branch destination address are addresses of the subroutines, storing the branch source address and the branch destination address as calling relationship information indicating a calling source subroutine, and a calling destination subroutine in said storage unit; and storing at least one of the cumulative execution time and the times of calling in the calling destination subroutine in the call source subroutine, as the profile corresponding to the calling relationship information, in the storage unit.

39. (PREVIOUSLY PRESENTED) A method according to claim 34, further comprising setting an execution environment of a source of the interrupt so as to generate the interrupt when the branch instruction is executed during the execution of the program.

40. (CURRENTLY AMENDED) An apparatus according to claim 13, wherein the collecting section generates a control table corresponding to each executor of the subroutine on the storage unit,

wherein the control table includes an executor managing table, a subroutine managing table, and a calling managing table,

wherein the executor managing table stores an identifier of the executor and a pointer to assign the subroutine managing table,

wherein the subroutine managing table is generated for every subroutine executed by the executor, the subroutine managing table storing a subroutine address, times of calling of the subroutine, a cumulative execution time of the subroutine, the last called time of the subroutine, and a pointer to assign the calling managing table, and

wherein the calling managing table is generated for every subroutine called by the subroutine, the calling managing table storing a branch source address as a calling subroutine address, a branch destination address as a called subroutine address, times of calling of the called subroutine, a cumulative execution time of the called subroutine, the last called time of the called subroutine, and a pointer to specify the subroutine managing table managing the calling subroutine.

41. (CURRENTLY AMENDED) A computer readable medium according to claim 23,

wherein the computer further executes the process by generating a control table corresponding to each executor of the subroutine on the storage unit,

wherein the control table includes an executor managing table, a subroutine managing table, and a calling managing table,

wherein the executor managing tables stores an identifier of the executor and a pointer to assign the subroutine managing table,

wherein the subroutine managing table is generated for every subroutine executed by the executor, the subroutine managing table storing a subroutine address, times of calling of the subroutine, a cumulative execution time of the subroutine, the last called time of the subroutine, and a pointer to assign the calling managing table,

wherein the calling managing table is generated for every subroutine called by the subroutine, the calling managing table storing a branch source address as a calling subroutine address, a branch destination address as a called subroutine address, times of calling of the called subroutine, a cumulative execution time of the called subroutine, the last called time of the called subroutine, and a pointer to specify the subroutine managing table managing the calling subroutine.

42. (CURRENTLY AMENDED) A method according to claim 38, wherein the method further comprises generating a control table corresponding to each executor of the subroutine on the storage unit,

wherein the control table includes an executor managing table, a subroutine managing table, and a calling managing table,

wherein the executor managing table stores an identifier of the executor and a pointer to assign the subroutine managing table,

wherein the subroutine managing table is generated for every subroutine executed by the executor, the subroutine managing table storing a subroutine address, times of calling of the subroutine, a cumulative execution time of the subroutine, the last called time of the subroutine, and a pointer to assign the calling managing table, and

wherein the calling managing table is generated for every subroutine called by the subroutine, the calling managing table storing a branch source address as a calling subroutine address, a branch destination address as a called subroutine address, times of calling of the called subroutine, a cumulative execution time of the called subroutine, the last called time of the called subroutine, and a pointer to specify the subroutine managing table managing the calling subroutine.